

Optical Monte Carlo in LarSoft : Update 14 April 2010

Ben Jones, MIT

Since Last Time:

Finished OpticalMCOutput Class

Mostly useful for debugging, building per-photon Ttrees at global scope from several modules on-the-fly

Added LarG4Parameters Class

To transfer config dependent parameters from LarG4 to other modules. Singleton class which takes data from the config selected by jobcontrol and makes them accessible at LArG4 scope

Updated MaterialPropertyLoader Class

Now can specify different momentum points for each material property to be loaded

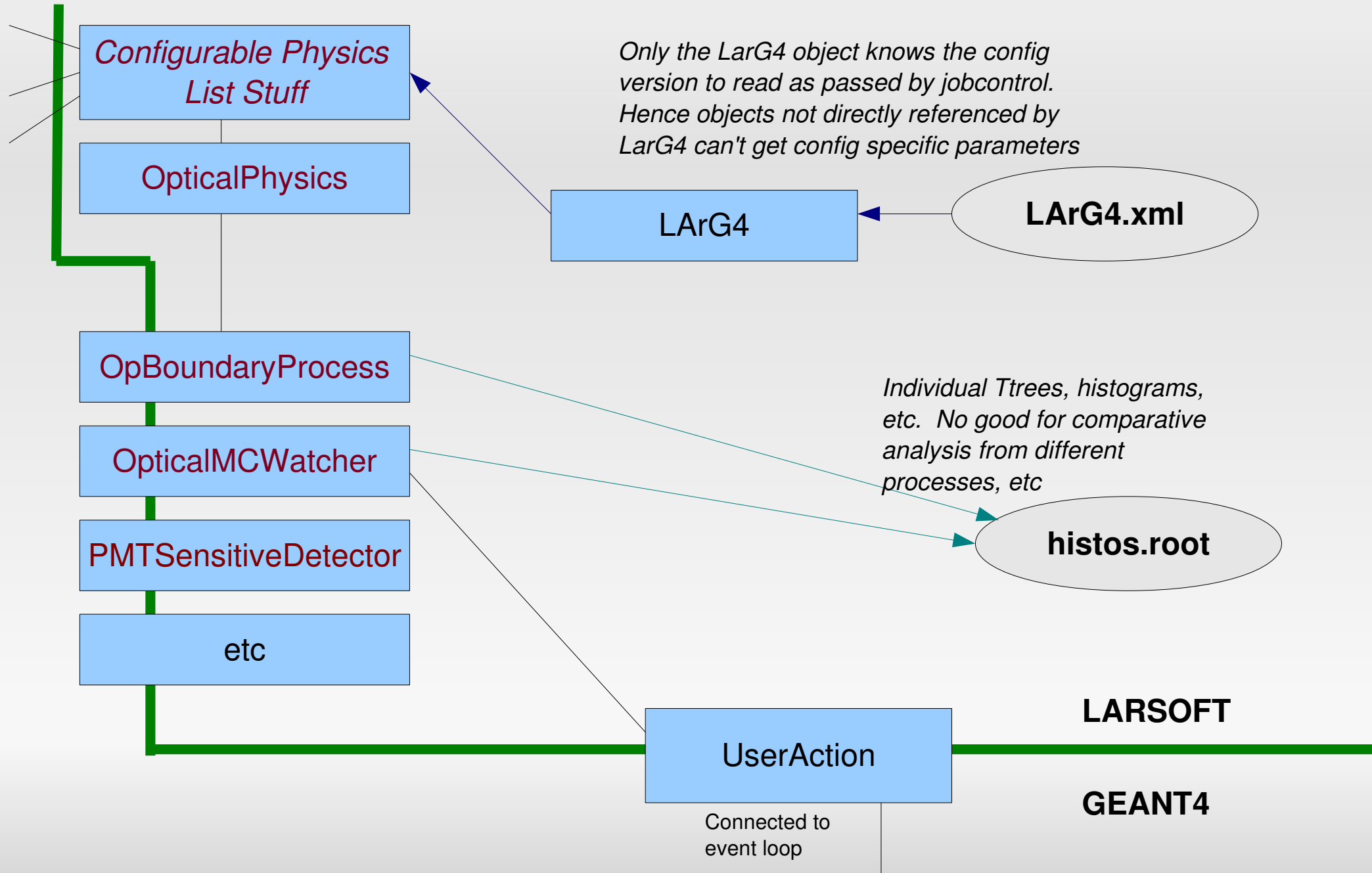
Worked on Putting Optical Components into the Simulation

Most of this talk...

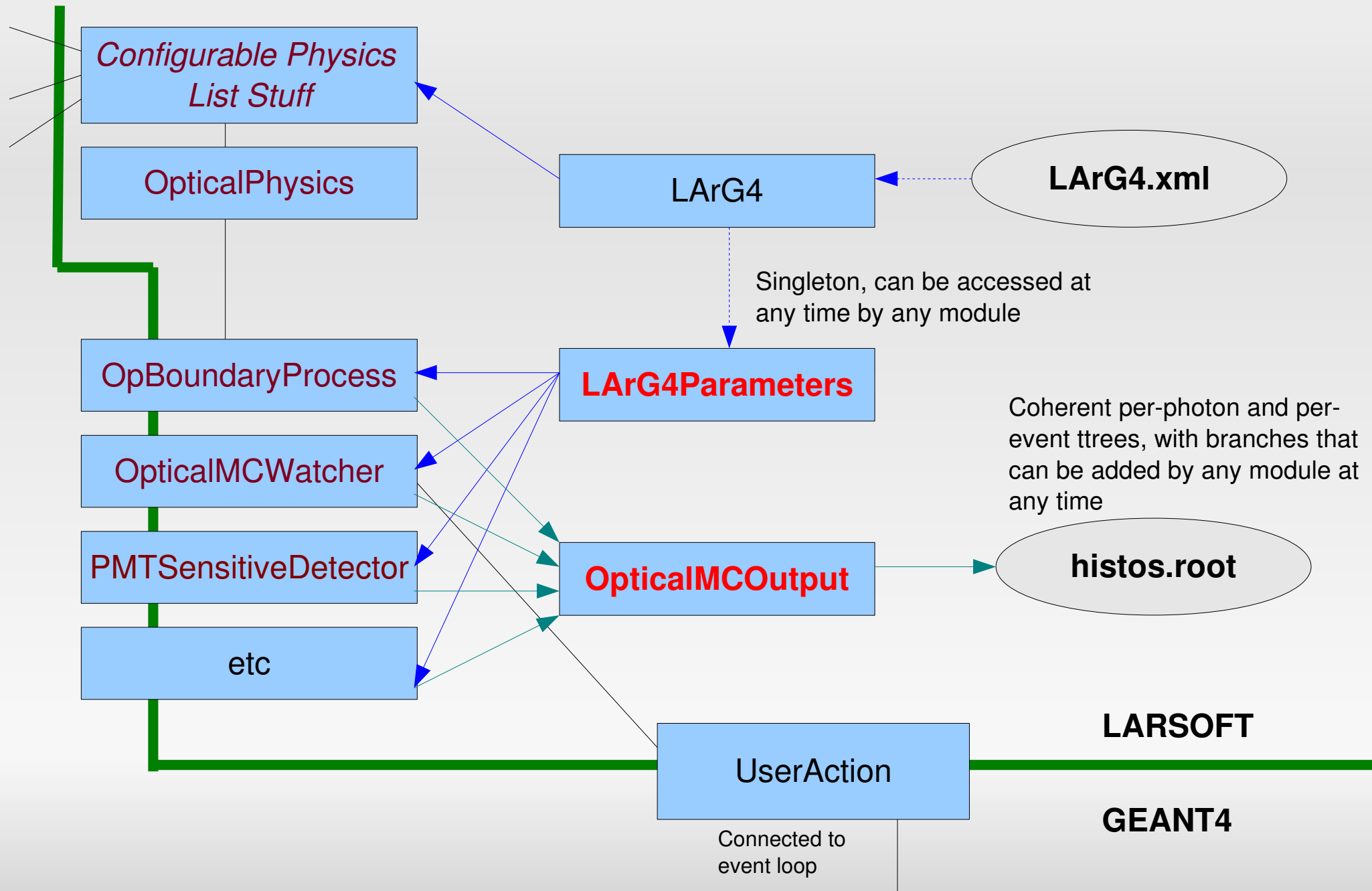
Daydreamed about and discussed XMLMaterialPropertyLoader

But not started implementation yet

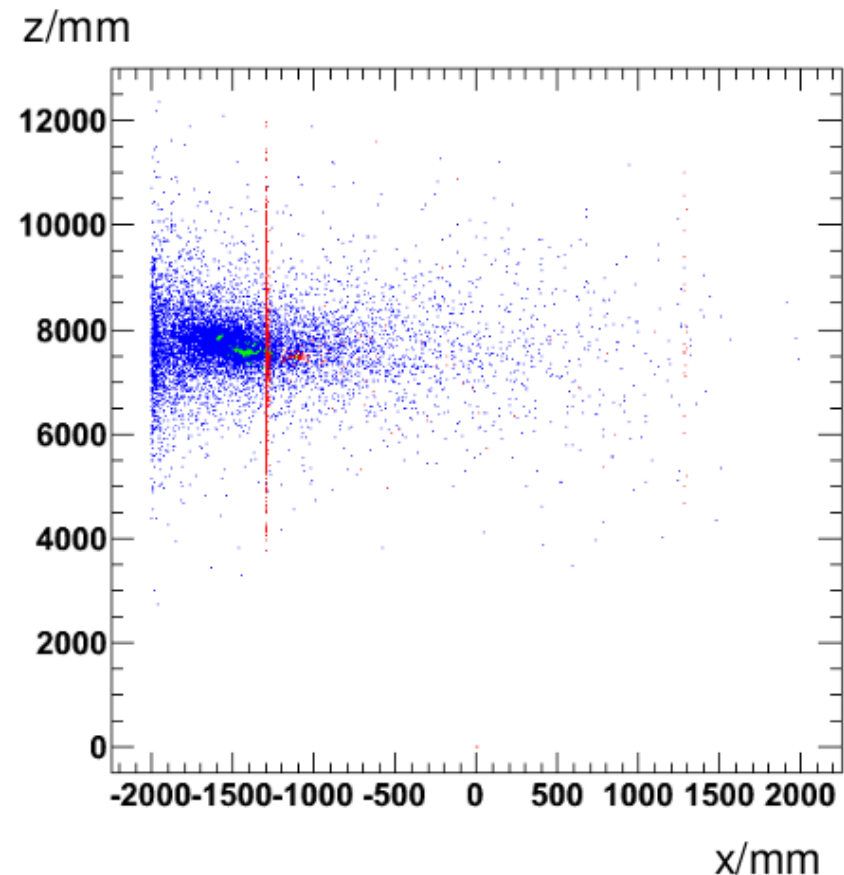
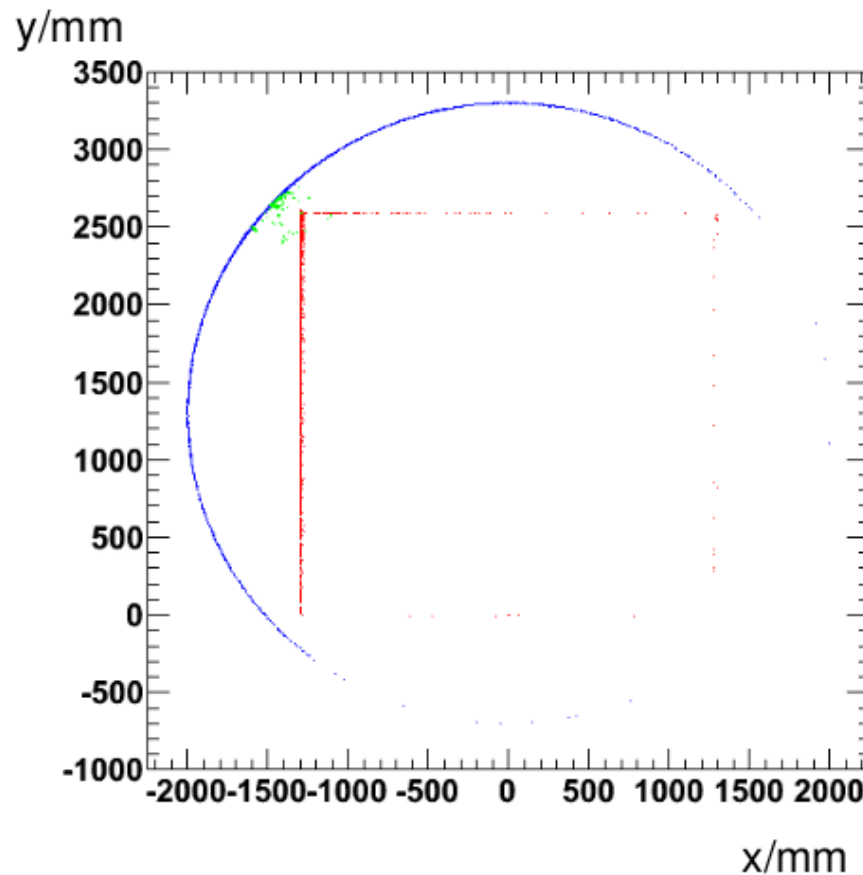
OpticalMCOutput and LArG4Parameters



OpticalMCOutput and LArG4Parameters



A Pretty Picture From OpticalMCOutput



Photons produced in Lar

Photons absorbed at surface with known properties

Photons absorbed at surface with unknown properties

Optical Components

- Broadly speaking, 4 optical components may important for microboone. In order of decreasing importance:

PMT's, Wavelength Shifting Plates,

Underway

Reflectors, Waveguides

Not yet started,
but easy
(hopefully)

And for each component, need to

- **1) Define optical properties, physics processes, hit recording / sensitivity, etc of each component**
- **2) Insert relevant volumes into the geometry in realistic positions**

Underway

Not sure how to
approach this...

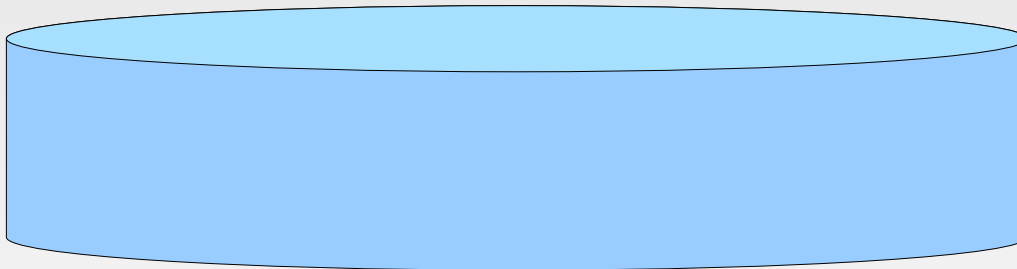
PMTs

For testing purposes, the following geometry is used to represent a super-simplified PMT:



TPB Plate

A volume with material "TPB"

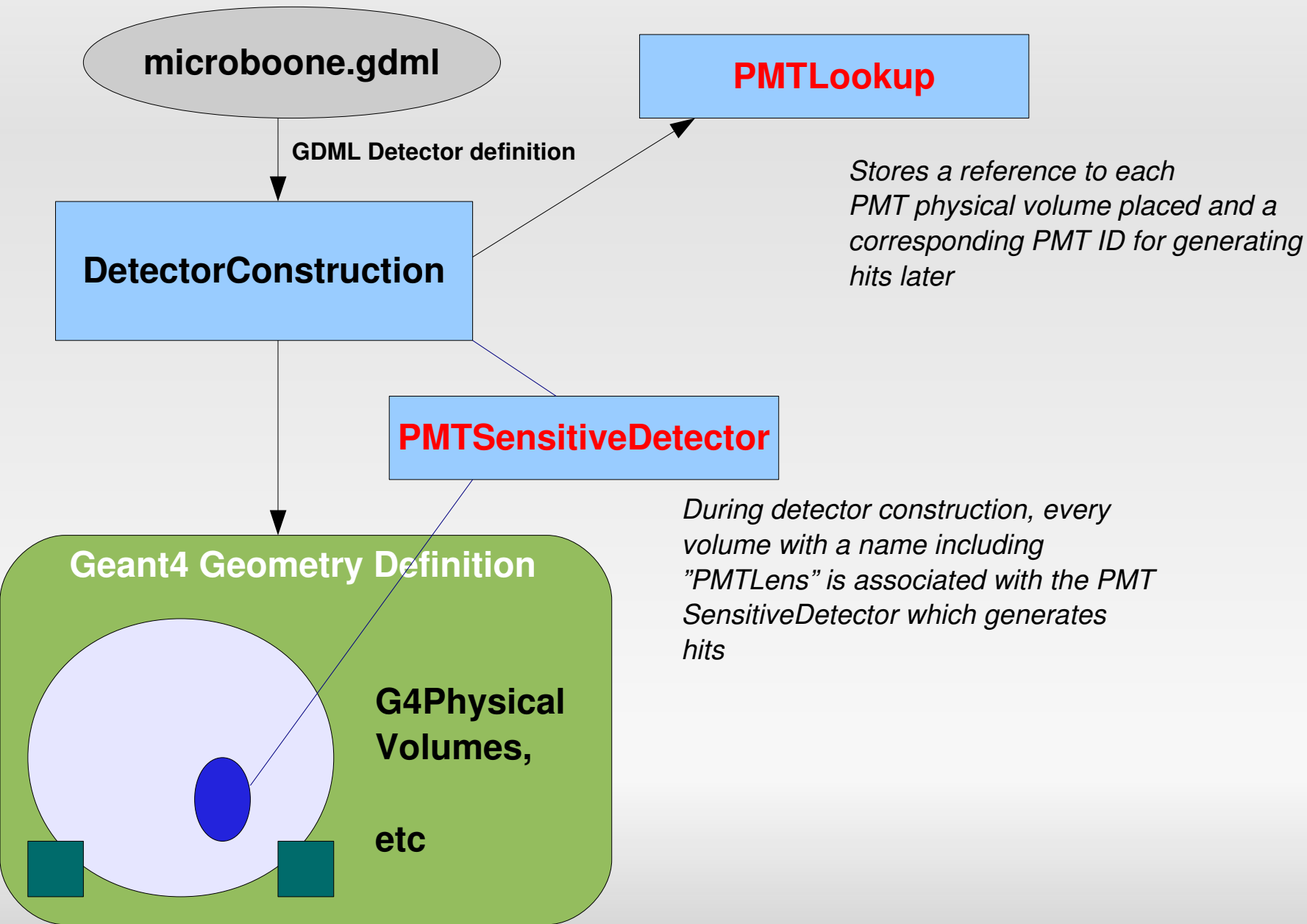


PMTLens

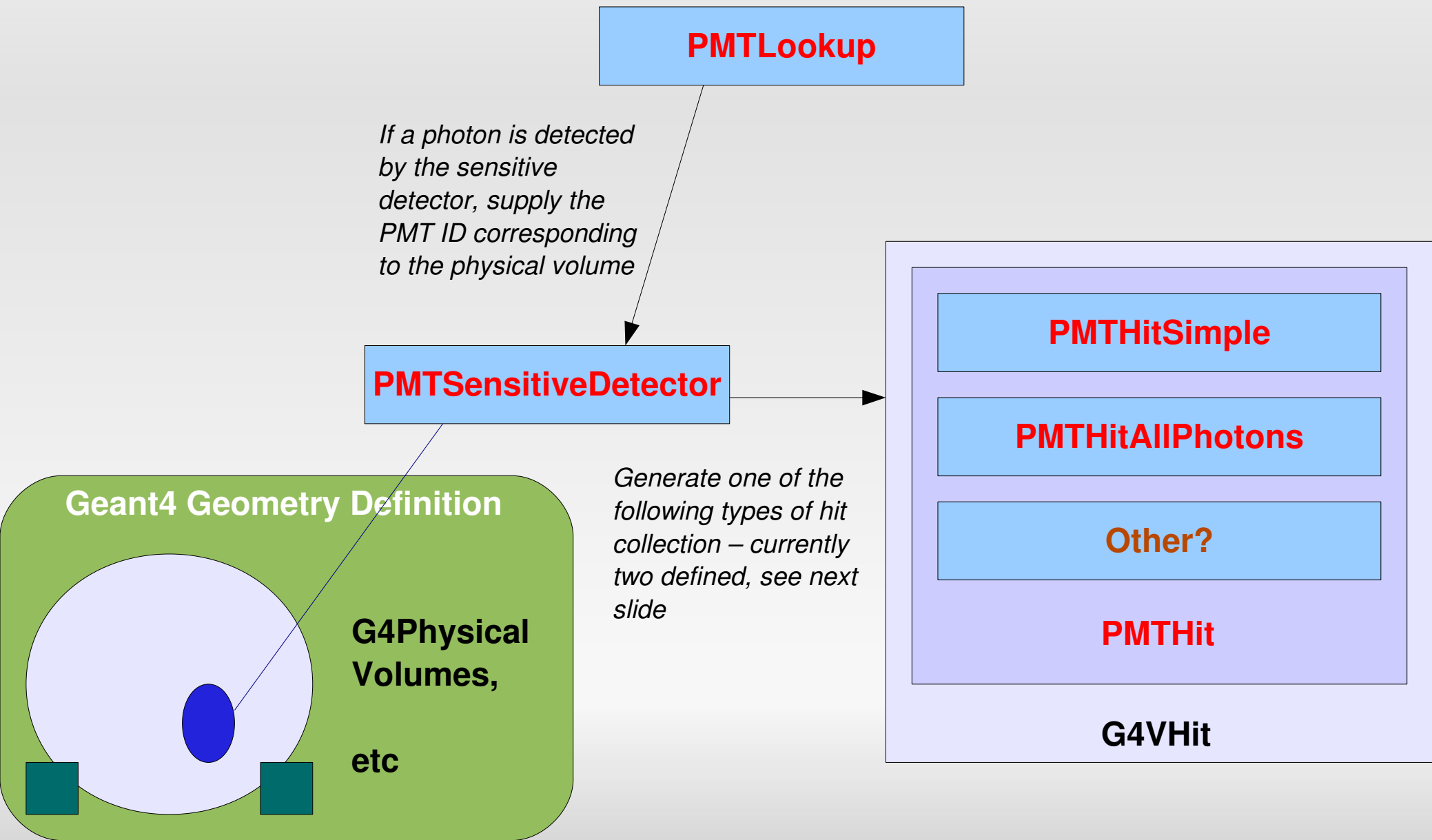
A volume with name containing "PMTLens", which is associated with a sensitive detector in the DetectorConstruction module

Currently use cylindrical geometries, but the shape doesn't matter to the implementation

PMT's – Detector Construction Time



PMT's – Simulation Time



PMTHit Definitions

- The PMTSensitiveDetector produces a collection of PMTHits to store in the event, one PMTHit per PMT
- Currently, the only necessary method of PMTHit is AddPhoton. This accepts photon momentum, position and trackID and uses it to store some kind of local data inside the hit
- There are currently two implementations of PMTHit, more can be defined in the future
- At the moment, the sensitive detector produces one of the two. Any call for generating more than one set?

PMTHitSimple

Stores:

Summed energy of photons
Number of photons detected

Optionally over a threshold

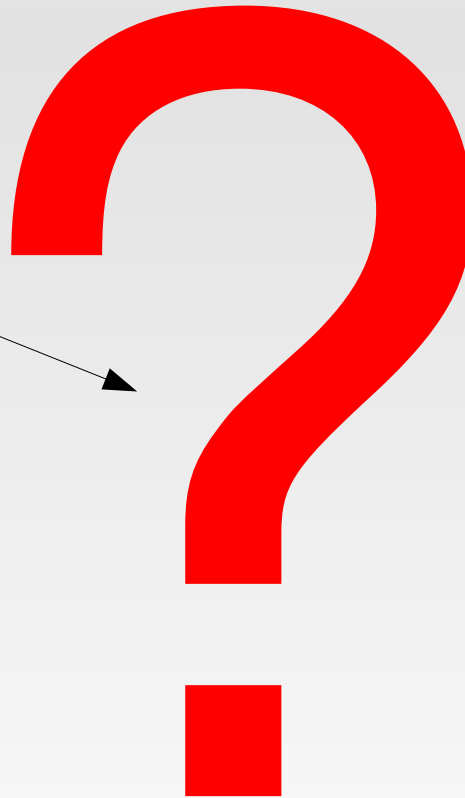
PMTHitAllPhotons

Stores:

std::vector of a struct containing
4-momentum of incident photon
4-position of incident photon
trackID of incident photon

Digitisation / pulse shape

Schematic of my current
knowledge of how to do this
part of the simulation

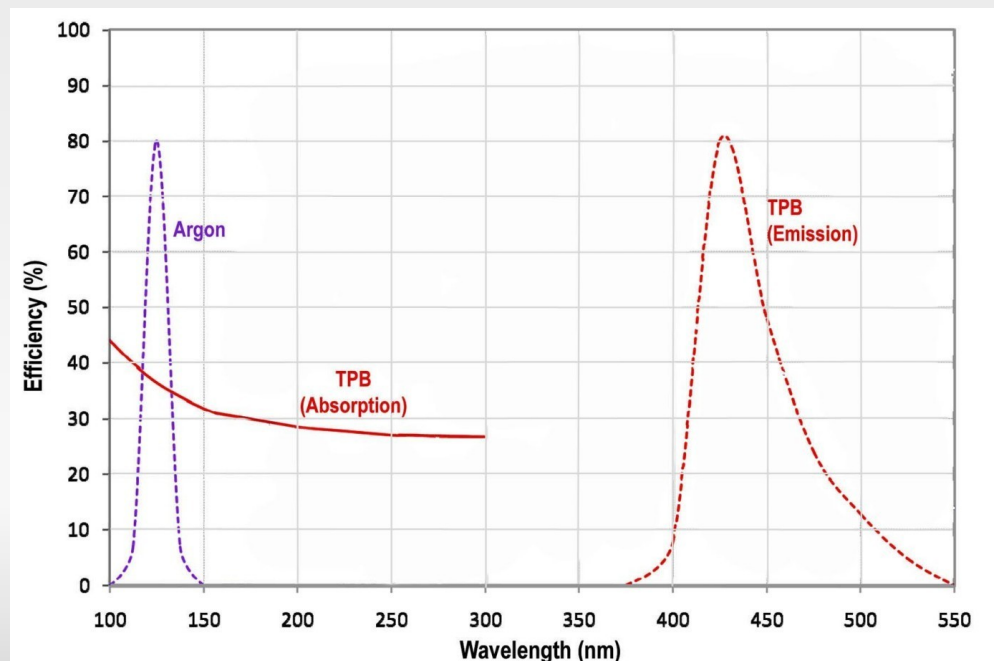


TPBPlate

- In the **gdml file**, defined material **TPB**. Defined volumes of this material in the geometry definition (called **TPBPlate1, TPBPlate2, etc**)
- In **OpticalPhysics physics list**, added process **G4OpWLS**
- In **DummyMaterialPropertyLoader**, added ***absorption spectrum, emission spectrum, time constant*** and ***mean photon yield*** for the wavelength shifting process in TPB

As with all optical processes, finding the correct parameterisations to use is an ongoing project

As yet I haven't fully tested this process – trying to get output from PMT's first



GDML, Defining geometry

- Now most physics processes and optical components are in place, we need to place them in correct spots in the geometry.
- How??

microboone.gdml

```
<tube name="PMTLens"
  rmax="(100)"
  z="(5)"
  deltaphi="2*(3.1415926535897)"
  aunit="rad"
  lunit="cm"/>
<tube name="TPBPlate"
  rmax="(100)"
  z="(1)"
  deltaphi="2*(3.1415926535897)"
  aunit="rad"
  lunit="cm"/>
<tube name="TPCWireVert"
  rmax="0.5*(0.015)"
  z="(256)"
  deltaphi="2*(3.1415926535897)"
  aunit="rad"
  lunit="cm"/>
<box name="TPCPlaneVert"
  x="(0.15*(2.54))"
  y="(256)"
  z="(1200)"
  lunit="cm"/>
<tube name="TPCWire0"
...
...
```

```
<flxi09.fnal.gov> wc microboone.gdml -l
41593 microboone.gdml
```

XMLMaterialPropertyLoader

- This has been a feature of EVERY talk I've given on LarSoft. Usually I say I'm working on it. I'm actually not working on it.
- This is because using the FMWK xml parsing scheme, it will be very tedious, and right now the return seems limited.
- A similar problem was solved for KATRIN by developing a new xml parsing method. I would like to implement this method in LarSoft using a library written by Dan Furse at MIT.
- In this new scheme, clever templating is used to load a predefined xml schema into a heirachy of C++ objects which directly corresponds to that schema. This makes the task trivial.
- The library is ready to go, and could be used easily in LarSoft.

Templated xml parsing

My materials xml file

```
<materialdoc>
  <material name = "LAr">
    <varproperty name = "RAYLEIGH" unit = "cm">
      Rayleigh scattering length in liquid argon
      <value momentum = 9.5 > 90 </value>
      <value momentum = 9.7 > 91 </value>
      <value momentum = 9.9 > 92 </value>
    </property>
    <constproperty name = "FASTTIMECONSTANT" unit = "ns">
      Fast time constant for scintillation in liquid argon
      <value> 6 </value>
    </property>
  </material>
</materialdoc>
```

**My schema definition,
in text file**

**Clever templating,
c/o xml parser**

C++ Object heirachy in LArSoft.

```
materialdoc->get_material("LAr")->get_property("RAYLEIGH")->get_value(9.7)
```

*** -this is neither real xml nor the objects will really look. But it gives an idea.**

Summary

- Since the last time I presented, there has been a lot of progress on OpticalMC
- Can generate per photon or per event trees with OpticalMCOutput for easy debugging / analysis
- Can pass parameters around LarG4 with LarG4Parameters
- PMT sensitive detector defined, which creates and stores PMTHits
- Two types of PMTHit defined, still more flexibility allowed
- TPB plate processes attached to TPB material and optical physics list
- A plan of attack in place for XMLMaterialPropertyLoader
- Still lots of work to do.